Appendix A.
Reference Summary

Topic: Ignore

This appendix provides a summary of the reference items defined in the Scheme Support Component. Just the name and a brief description of each item is given. There is a separate section for each type of item (class, function, Scheme extension, test harness command, etc.).

Classes

Topic: Ignore

param_string	References parameters.
scheme_error_callback_list	Provides a type–safe interface to the toolkit_callback_list and the execute method.

Functions

Topic: Ignore

active_part_context	Gets the active PART_CONTEXT. Use this method when adding a new ENTITY to a PART.
api_pm_add_entity	Adds an ENTITY to a PART.
api_pm_create_part	Creates a new PART.
api_pm_delete_all_states	Deletes all states.
api_pm_delete_part	Deletes a PART.
api_pm_entity_id	Gets the entity ID and part for an ENTITY.
api_pm_load_part	Loads a file into a PART.
api_pm_lookup_entity	Gets an entity given an ID and a PART.

api_pm_name_state . . . . . Names the current state.

api_pm_note_state . . . . . Marks the end of a state.

api_pm_part_entities . . . . . Gets a list of entities in a PART.

api_pm_remove_entity . . . . . Removes an ENTITY from a part.

api_pm_roll_n_states . . . . . Rolls forward or backward a specified number of states.

api_pm_roll_to_state . . . . . Rolls to the start of a named state.

api_pm_save_part . . . . . Saves a PART to a file.

api_pm_start_state . . . . . Marks the start of a state.

delete_GC_Objects . . . . . Deletes the GC_Object.

get_part_context . . . . . Gets the PART_CONTEXT from an ENTITY.

get_scheme_error_callback_list . . . . . Gets a global list of scheme error callbacks.

get_Scm_String . . . . . Creates a C++ const char* from a Scheme string object.

is_Scm_Real_List . . . . . Determines if a Scheme object is a list of reals.

refresh_all . . . . . Refreshes all views.

SchemeEvaluate . . . . . Evaluates a string or Scheme object.

SchemeLoad . . . . . Loads a Scheme file into memory.

scheme_process . . . . . Builds and evaluates a Scheme command, optionally echoing the prompt and result.

start_entity_creation . . . . . Prepares for the definition of a new ENTITY.

start_entity_modification . . . . . Prepares for ENTITY modification.

Options

Topic: Ignore

full_error_msg . . . . . Controls the amount of information issues for ACIS error messages.

journal_result_limit . . . . . Sets the maximum number of characters used to journal the result of a Scheme operation when journaling is turned on.

save_version . . . . . Sets the ACIS version to be used by the part:save Scheme extension.

Scheme Extensions

Topic: Ignore

check–interrupt . . . . . Executes process_event to process pending events if interrupts are enabled.

dl–item:color . . . . . Gets the color of a display item in the display list.

dl–item:display . . . . . Adds a display item to the display list for all views.

dl–item:erase . . . . . Removes a display item from the display list.

dl–item:point . . . . . Creates a point item for the display list.

dl–item:polyline . . . . . Creates a polyline item for the display list.

dl–item:set–color . . . . . Sets the color of a display item.

dl–item:text . . . . . Creates a text display list item.

dl–item? . . . . . Determines if a Scheme object is a display item.

edge:chain . . . . . Gets a chain of edges given one or two edges.

entity:color . . . . . Gets the display color for an entity.

entity:display . . . . . Displays entities.

entity:displayed? . . . . . Determines if a Scheme entity is displayed.

entity:erase . . . . . Erases but does not remove the specified entity or list of entities.

entity:highlighted? . . . . . Determines if an entity is highlighted.

entity:part . . . . . Gets the part to which an entity belongs.

entity:select . . . . . Selects an entity or list of entities.

entity:selected? . . . . . Determines if an entity is selected.

entity:set–color . . . . . Sets the display color for an entity or list of entities.

entity:set–highlight . . . . . Sets highlighting for an entity or list of entities.

entity:set–part . . . . . Sets the part to which an entity belongs.

entity:set–render . . . . . Marks whether or not an entity is to be rendered.

entray:position . . . . . Gets the position of the vertex of the entity component of an entray that is closest to the ray.

entray:vertex . . . . . Gets the vertex on the entity component of the entray that is closest to the ray.

env:active–part . . . . . Gets the active part.

env:active–view . . . . . Gets the currently-active view.

env:active–wcs–color . . . . . Gets the display color of the active WCS.

env:auto–display . . . . . Gets the auto–display mode.

env:count–parts . . . . . Gets the number of defined parts.

env:default–color . . . . . Gets the current default color value setting.

env:highlight–color . . . . . Gets the highlight color.

env:parts . . . . . Gets a list of all parts in the environment.

env:point–size . . . . . Gets the display size of a point.

env:point–style . . . . . Gets the display style of a point.

env:set–active–part . . . . . Sets the active part.

env:set–active–view . . . . . Sets the specified view as the active view.

env:set–active–wcs–color . . . . . Sets the color of the active WCS.

env:set–auto–display . . . . . Sets the automatic display of newly–created entities on or off.

env:set–default–color . . . . . Sets the default color for newly–created entities.

env:set–highlight–color . . . . . Sets the highlight color.

env:set–point–size . . . . . Sets the displayed size of a point.

env:set–point–style . . . . . Sets the display style for a point.

event . . . . . Creates a pick–event.

event:alt? . . . . . Determines if the Alt key is pressed in a pick–event.

event:button . . . . . Gets the button from a pick–event.

event:control? . . . . . Determines if the Ctrl key is pressed in a pick–event.

event:left? . . . . . Determines if the left mouse button is pressed in a pick–event.

event:middle? . . . . . Determines if the middle mouse button is pressed in a pick–event.

event:ray . . . . . Gets the pick ray from a pick–event.

event:right? . . . . . Determines if the right mouse button is pressed in a pick–event.

event:shift? . . . . . Determines if the Shift key is pressed in a pick–event.

event:view . . . . . Gets the view from a pick–event.

event:x . . . . . Gets the *x*–coordinate from a pick–event.

event:y . . . . . Gets the *y*–coordinate from a pick–event.

event? . . . . . Determines if a Scheme object is a pick–event.

filter:color . . . . . Creates an entity–filter that tests on the basis of color.

filter:display . . . . . Creates a filter entity that tests on the basis of a display status.

gvector:view . . . . . Creates a gvector in the view coordinate system.

history . . . . . Gets a history stream from an entity, part, ID, or string.

history:get–modified–faces . . . . . Finds faces that have been deleted, created, or modified since a particular state.

history:load . . . . . Loads one or more history streams and associated entities from a file.

history:save . . . . . Saves all entities, entity–id, and roll information associated with a history stream to a file.

history:size . . . . . Returns the size of the given history stream.

journal:abort . . . . . Terminates the journal single stepping process.

journal:append . . . . . Opens an existing journal file and appends additional journal data to the end of the file.

journal:load . . . . . Loads a journal file one line at a time, journaling each command as it is executed.

journal:off . . . . . Closes the current journal file and turns off journaling.

journal:on . . . . . Closes the current journal file and opens a new journal file.

journal:pause . . . . . Disables journaling temporarily but leaves the journal file open.

journal:resume . . . . . Resumes journaling in the journal file again after pausing.

journal:save . . . . . Saves the current journal to a file.

journal:step . . . . . Enables or disables single stepping of journal file on and off.

load–dll . . . . . Loads a dynamically–linked library for the NT platform.

part . . . . . Gets a Scheme part from its identification number.

part:clear . . . . . Deletes all entities from the active part.

part:clear–selection . . . . . Clear the list of currently selected entities.

part:close . . . . . Closes (deletes) a part.

part:debug . . . . . Gets information about the entities in a part and writes it to the active output port.

part:delete–rendering–context . . . . . Deletes a rendering context from a part.

part:entities . . . . . Gets a list of all top–level entities in a part.

part:get–distribution–mode . . . . . Returns whether distributed history is on or off.

part:load . . . . . Loads a part from a file into the active part.

part:modified? . . . . . Determines if a part has been modified.

part:name . . . . . Gets the name of a part.

part:new . . . . . Creates a new part.

part:save . . . . . Saves all entities in a part to a file.

part:save–selection . . . . . Saves a list of entities to a file.

part:selection . . . . . Returns a list of currently selected entities.

part:set–distribution–mode . . . . . Turns distributed history on or off.

part:set–name . . . . . Sets the name of a part.

part:views . . . . . Gets all of the views displaying a part.

part? . . . . . Determines if a Scheme object is a part.

pick:aperture . . . . . Gets the dimensions of a pick aperture.

pick:edge . . . . . Gets an edge from a pick event.

pick:entity . . . . . Gets an entity from the pick event.

pick:face . . . . . Gets a face from a pick event.

pick:in–region . . . . . Gets all entities in a region specified by two pick events.

pick:position . . . . . Gets the position of a pick event.

pick:ray . . . . . Gets a ray from a pick event.

pick:set–aperture . . . . . Sets the pick aperture in x and y.

pick:vertex . . . . . Gets a vertex from a pick event.

position:view . . . . . Creates a new position given coordinates x, y, and z in the view coordinate system.

read–event . . . . . Creates a pick–event Scheme object from mouse input.

ro:color . . . . . Gets the color of a display item in the display list.

ro:delete . . . . . Deletes the rendering object.

ro:display . . . . . Displays the rendering object.

ro:erase . . . . . Erases the rendering object.

ro:line . . . . . Creates a line.

ro:new . . . . . Creates a new rendering object.

ro:point . . . . . Creates a point item for the display list.

ro:polyline . . . . . Creates a polyline item for the display list.

ro:text . . . . . Creates a text display list item.

roll . . . . . Rolls to a previous or later state.

roll:back? . . . . . Determines if there are any previous states to roll back to.

roll:debug . . . . . Writes information about the rollback operations to the debug file.

roll:delete–all–states . . . . . Deletes all delta states.

roll:delete–following–states . . . . . Deletes all delta states after the current delta state.

roll:delete–inactive–states . . . . . Deletes all delta states after the current one.

roll:delete–previous–states . . . . . Deletes all delta states before the current one.

roll:forward? . . . . . Determines if there are any states following which can be rolled forward to.

roll:get–logging . . . . . Returns the state of the internal logging flag.

roll:mark–end . . . . . Marks the end of a block of functions for rolling.

roll:mark–start . . . . . Marks the start of a block of functions for rolling.

roll:merge–delta–state . . . . . Merges one delta–state with an adjacent one.

roll:merge–delta–states . . . . . Merges a range of delta states.

roll:name–state . . . . . Attaches a name to the current state for rolling.

roll:named–states . . . . . Gets a list of states matching a wildcard name.

roll:set–logging . . . . . Sets the global logging flag.

roll:set–max–states . . . . . Sets the maximum number of delta states to keep.

roll:to–state–id . . . . . Rolls to a delta state specified by its integer identifier.

system:bell . . . . . Sounds the system bell at a specified frequency for a specified duration.

system:command . . . . . Executes a system command.

system:getenv . . . . . Gets the value of an environment variable.

system:name . . . . . Gets a string specifying the hardware environment.

system:play–sound . . . . . Plays a .wav file on Windows.

system:set–timer–off . . . . . Sets the timer off.

system:set–timer–on . . . . . Sets the timer on.

system:sleep . . . . . Freezes the view for a specified amount of time.

system:time–string . . . . . Gets a string specifying the current system time and date.

ui:error–dialog . . . . . Displays a message string in an error dialog box.

ui:info–dialog . . . . . Displays a message string in an information dialog box.

ui:prompt . . . . . Displays a message string on the command line.

ui:warning–dialog . . . . . Displays a message string in a warning dialog box.

ui:yesno–dialog . . . . . Displays a message string in a yes/no dialog box.

versionid . . . . . Gets the current version ID of the ACIS executable.

view:bg–color . . . . . Gets a view's background color.

view:clear . . . . . Clears the contents of a view.

view:delete . . . . . Deletes a view.

view:display–facets . . . . . Displays the facets.

view:display–param–lines . . . . . Displays parameter lines.

view:display–surface–polys . . . . . Displays surface polygons.

view:draw–point . . . . . Draws a temporary point.

view:draw–polyline . . . . . Draws a temporary polyline.

view:draw–text . . . . . Draws a temporary text string.

view:eye . . . . . Gets the eye position of a view.

view:fg–color . . . . . Gets a view's foreground color.

view:handle . . . . . Gets a handle of the specified view.

view:height . . . . . Gets the height of a view.

view:hither . . . . . Gets the hither clipping distance of a view.

view:out . . . . . Gets the vector from the target to the eye position.

view:perspective? . . . . . Determines if perspective is on for a view.

view:right . . . . . Gets the right vector of a view.

view:set . . . . . Sets a view's eye position, target position, and up vector.

view:set–bg–color . . . . . Sets a view's background color.

view:set–clipping . . . . . Sets the hither and yon clip distances.

view:set–draw–mode . . . . . Sets the drawing mode for temporary graphics in a view.

view:set–eye . . . . . Sets a view's eye position.

view:set–fg–color . . . . . Sets the foreground color for a view.

view:set–font . . . . . Sets the text font for a view.

view:set–font–size . . . . . Sets the text font size for a view.

view:set–hither . . . . . Sets a view's hither clip distance.

view:set–line–style . . . . . Sets the line style for temporary lines.

view:set–line–width . . . . . Sets the width of temporary lines in a view created with view:draw–polyline.

view:set–perspective . . . . . Sets the perspective or orthographic modes.

view:set–point–size . . . . . Sets the size of temporary points in a view.

view:set–point–style . . . . . Sets the style for temporary points in a view created with view:draw–point.

view:set–rb–mode . . . . . Sets the rubberbanding mode for a view.

view:set–size . . . . . Sets the width and height of the view window in world coordinates.

view:set–target . . . . . Sets a view's target position.

view:set–title . . . . . Sets the banner name for a specified view.

view:set–up . . . . . Sets the up vector for a view.

view:set–viewport . . . . . Sets the viewport of a view.

view:set–yon . . . . . Sets the yon clip distance of a view.

view:target . . . . . Gets the target position of a view.

view:up . . . . . Gets the up vector of a view.

view:viewport . . . . . Gets the viewport origin, width, and height of a view.

view:width . . . . . Gets the width of a view.

view:with–handle . . . . . Gets the view of the specified handle.

view:yon . . . . . Gets the yon clip distance of a view.

view? . . . . . Determines if a Scheme object is a view.

void? . . . . . Determines if a Scheme object is void.

Typedefs

Topic: Ignore

entity_id_t . . . . . Identifier for an ENTITY.

ScmObject . . . . . Declaration of Scheme object data type.